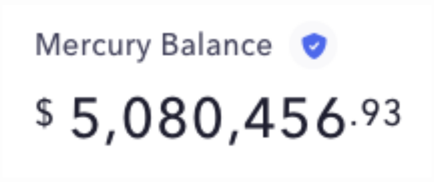
Here we go again. I started another company. The money is in the bank.


Mercury Balance

$ 5,080,456.93

What is the tiny corp?

The tiny corp is a computer company. We sell computers for more than they cost to make; I've been thinking about this one for a while. In the limit, it's a chip company, but there's a lot of intermediates along the way.

The human brain has about 20 PFLOPS of compute. I've written various blog posts about this. Sadly, 20 PFLOPS of compute is not accessible to most people, costing about $1M to buy or $100/hr to rent.

With the way AI is going, we risk large entities controlling the majority of the compute in the world. I do not want "I think there's a world market for maybe five computers." to ever be the world we live in.

The goal of the tiny corp is: **"to commoditize the petaflop"**

What is tinygrad?

I started tinygrad in Oct 2020. It started as a toy project to teach me about neural networks, it's now carved out a good niche in the inference space running the model in openpilot, and soon will be a serious competitor to PyTorch in many places.

The main advantage is in the tinygrad IR. It has 12 operations, all of which are ADD/MUL only. `x[3]` is supported, `x[y]` is not. Matrix multiplies and convolutions are just multiplies and sums, surrounded by a bunch of zero cost movement operations (like reshape, permute, expand).

```
# a fast matmul in tinygrad (a@b works also of course)
from tinygrad.tensor import Tensor
N = 2048; a, b = Tensor.randn(N,N), Tensor.randn(N,N)
c = (a.reshape(N,1,N) * b.permute(1,0).reshape(1,N,N)).sum(axis=2)
```

tinygrad is lazy, like Haskell, to allow op fusion without the user ever having to think about it.

Ok, so?

The current crop of AI chip companies failed. Many of them managed to tape out chips, some of those chips even worked. But not a single one wrote a decent framework to use those chips. They had similar performance/$ to NVIDIA, and way worse software. Of course they failed. Everyone just bought stuff from NVIDIA.

I think the only way to start an AI chip company is to start with the software. The computing in ML is not general purpose computing. 95% of models in use today (including LLMs and image generation) have all their compute and memory accesses statically computable.

Unfortunately, this advantage is thrown away the minute you have something like CUDA in your stack. Once you are calling in to Turing complete kernels, you can no longer reason about their behavior. You fall back to caching, warp scheduling, and branch prediction.

tinygrad is a simple framework with a PyTorch like frontend that will take you all the way to the hardware, without allowing terrible Turing completeness to creep in.

The Red Team (AMD)

10 or so companies thought it was a good idea to tape out chips. Ironically, taping out the chip is the easy part. It requires a lot of capital, but that just involves convincing foolish investors that the space they are targeting is *SO HUGE* that even if they have a 3% chance of success it's worth investing. Investors fall for this, they invest, and the world tapes out useless AI chips.

There's a great chip already on the market. For $999, you get a 123 TFLOP card with 24 GB of 960 GB/s RAM. This is the best FLOPS per dollar today, and yet…nobody in ML uses it.

I promise it's better than the chip you taped out! It has 58B transistors on TSMC N5, and it's like the 20th generation chip made by the company, 3rd in this series. Why are you so arrogant that you think you can make a better chip? And then, if no one uses this one, why would they use yours?

So why does no one use it?

The software is terrible! There's kernel panics in the driver. You have to run a newer kernel than the Ubuntu default to make it remotely stable. I'm still not sure if the driver supports putting two cards in one machine, or if there's some poorly written global state. When I put the second card in and run an OpenCL program, half the time it kernel panics and you have to reboot.

That's the kernel space, the user space isn't better. The compiler is so bad that clpeak only gets half the max possible FLOPS. And clpeak is a completely contrived workload attempting to maximize FLOPS, never mind how many FLOPS you get on a real program (usually like 25%).

The software is called ROCm, it's open source, and supposedly it works with PyTorch. Though I've tried 3 times in the last couple years to build it, and every time it didn't build out of the box, I struggled to fix it, got it built, and it either segfaulted or returned the wrong answer. In comparison, I have probably built CUDA PyTorch 10 times and never had a single issue.

Where does the tiny corp come in?

Forget all that software. The RDNA3 Instruction Set is well documented. The hardware is great. We are going to write our own software.

If you were to tape out your own chip, you'd be struggling with both hardware bugs and software bugs, and you wouldn't be sure which one it is. Here, you have a good idea, and have the AMD provided driver as an open source reference.

This is life on easy mode, and I still doubt any of those AI startups could have done it. This is what the tiny corp is going to do to start. Build a framework, runtime, and driver for AMD chips.

AMD on MLPerf

Every couple months, MLCommons hosts MLPerf, a competition to train a common set of models fast. AMD has never been on the list, not because the hardware can't do it, but because the software can't. The list is dominated by NVIDIA, but Google TPUs, Intel CPUs, Intel Habana, Huawei Ascend, and Graphcore IPU have made appearances. Never seen AMD.

Our short term goal is to get AMD on MLPerf using the tinygrad framework.

But how do you make money doing that?

We don’t. We raised $5M from an amazing set of investors who are aligned with real value creation.

We make money selling computers for more than they cost to make. (preorder a tinybox today)

If we succeed at this project, we will be on the cutting edge of non NVIDIA AI compute. We have the ability to make the software, and that’s the hard part. Comparatively, taping out chips is easy. If we even have a 3% chance of dethroning NVIDIA and eating in to their 80% margins, we will be very very rich.

How can I help?

We’re hiring software engineers to work on tinygrad. In person in San Diego, looking for people who want to work hard and build something incredible. Come work on an open source project that, if done right, will play a role in the joint destiny of humanity and its machines.

I don’t want to live in a world of closed AI running in a cloud you’ve never seen, I want everyone to have an AI that they own, both training and inference. I want compute to be available from 50 different companies all competing to drive the price to zero. And I want an open source framework to run cutting edge AI on any one of those 50 chips as seamlessly as Linux supports 50 network cards.

If Elon has the FSD Chip and Dojo for Autopilot and Tesla’s robots, we have the tiny corp for openpilot and comma’s robots. comma, along with 100s of other companies, will need computers both big and small for training and inference. We will sell them those computers. If NVIDIA is the Apple, we are the Android.

There’s only one way to get hired at the tiny corp, and that’s by submitting high quality pull requests to tinygrad. Job interviews are obsolete, prove you can do the job by doing the job!

It should be fun for you, and if it isn’t, it’s probably not a good fit. It’s MIT licensed open source software, the work you do belongs to all of us, not like we are going to run off and sell IP or something. I hope you know me better than that. There’s bounties available if you are pushing forward on the main roadmap.

If you feel like you missed the beginning of comma, get in on this. The fun of the beginning is how much you get to shape.